A Multi-Team Multi-Semester Large-Scale Capstone Project Experience

Sean Owens[1], Tommy Morris[2], Gregory Hall[3]

Abstract – In 2010, NASA sponsored a Senior Design project to implement the ITU G.729 voice CODEC standard on an FPGA using Verilog HDL. This project is unique because it has allowed students to directly interact with NASA engineers. Another distinctive aspect of this project is that it has required four separate senior design teams to complete. Each team began the Senior Design course one semester after the previous team, and this staggered entrance forced inter-team cooperation and project management, and led students to move to an open-source development model. This paper will outline the key aspects of the project, explain how the multiple team structure has evolved over the four semesters while adhering to Mississippi State University curriculum requirements, and discuss the benefits and detriments associated with multiple team projects. It may serve as a reference model for larger, multiple-team capstone projects in the future.

Keywords: Capstone, Multi-team, CODEC

INTRODUCTION

The International Telecommunication Union's (ITU) G.729 audio encoding/decoding standard is a very popular coder/decoder (CODEC) for communications over low bandwidth systems. While software versions of the CODEC exist, specifically the ITU provided C implementation, NASA desired a solution that could minimize size, weight and power (SWAP). The goal of the capstone project presented in this paper was to accomplish these optimizations by designing and implementing the ITU G.729 standard using the Verilog hardware description language (HDL). Furthermore, NASA intends to use the results of this project to fuel research in the area of reconfigurable computing.

This capstone project is unique to the Mississippi State University Electrical and Computer Engineering (ECE) department because it has required four separate senior design teams to complete. The first team was tasked with designing the base system architecture; developing building blocks shared throughout the CODEC, and beginning implementation of the encoder subsystem. The second team completed implementation of the encoder module, developed system level validation process, and interfaced the encoder to a CPU. The third team architected the decoder module of the CODEC. Finally, the fourth team completed decoder implementation and developed a fully integrated CODEC prototype for demonstration to NASA. Each team began the Senior Design course one semester after the previous team, and this staggered entrance created many challenges for the individual team members. The main challenge faced was previous teams having to bring the new teams up to speed while still accomplishing their own goals. Other challenges were coordinating communications and work between teams. These challenges inherently provided situations for the teams to gain experience in inter-team cooperation, project management, and

---

[1] Mississippi State University, Box 911, Mississippi State, MS 39762, sas364@msstate.edu

[2] Mississippi State University, Box 9571, Mississippi State, MS 39762, morris@ece.msstate.edu

[3] NASA- Johnson Space Center, Mail Code- EV2, Houston, TX 77058, gregory.a.hall@nasa.gov

working in an open-source development environment. This paper will summarize the main objectives of the project, describe the evolution of the multi-team dynamic with respect to course requirements created for single team projects, and discuss the valuable experiences, as well as difficulties, inherent to multiple-team projects.

BACKGROUND

CODEC

The ITU G.729 CODEC is a high-intelligibility, low-bandwidth audio compression algorithm that encodes and decodes speech at 8 kbits/s using conjugate-structure algebraic code-excited linear prediction (CS-ACELP) [1]. The system takes in 16-bit Pulse Code Modulated (PCM) samples at 8 kHz. It collects a frame of 80 samples and then processes it into an encoded bit stream that is ready for transmission. After transmission, the bit stream is input into the decoder subsystem and is converted back into a digital audio signal. Figure 1 gives an illustration of the operation of the CODEC.

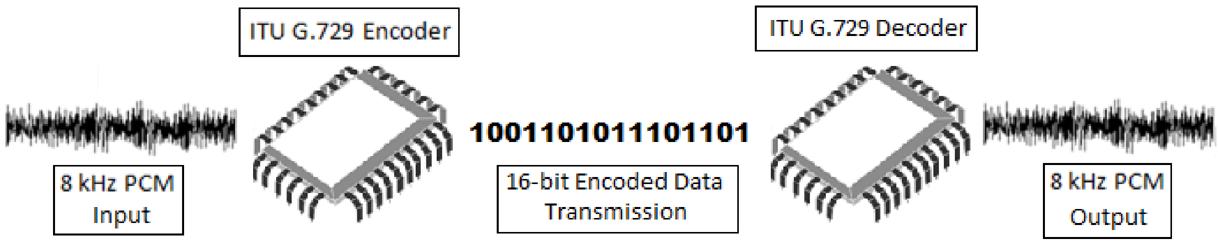

Figure 1: ITU G.729 CODEC Operation Illustration

Four Teams

Each of the four teams that worked on this project had unique goals and contributions to the overall product. The goals of the first team were to define the scope of the project, design a modular architecture that could support the CODEC, and establish a method for implementing the encoder. By defining the scope of the project, it was determined that the project proposed by NASA would require more than one team to complete. Designing the system architecture to be used for the project was the biggest challenge to the first team. An overview of the system is shown in Figure 2.

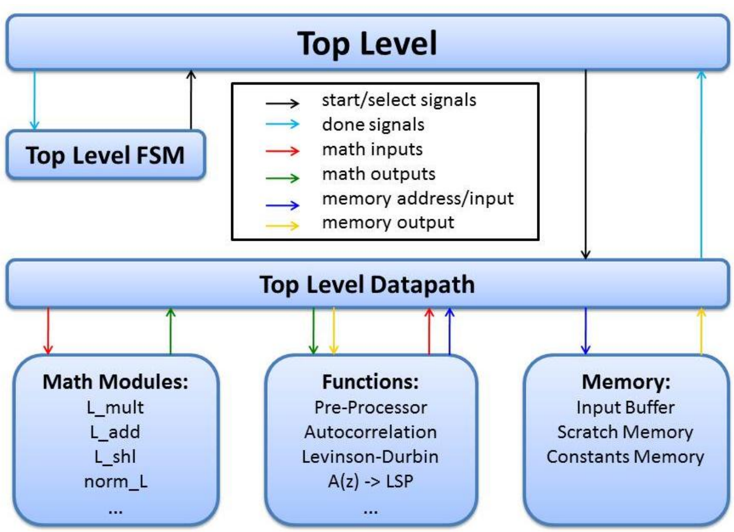

Figure 2: ITU G.729 Architecture

The encoding and decoding processes were broken down into functions that performed specific computations on the input frames. These functions used standardized math modules on data stored in the different memory blocks to complete the computations required. These subsystems communicated through a higher-level datapath, and the functions were executed according to the finite state machine (FSM) controller at the Top Level. Designing the architecture in this way allowed the members of the team to work on and test separate parts of the system at the same time without conflict. Finally, the first team developed a template for creating individual modules in the system that provided a coherency throughout the code. Using this template reduced the overhead time required to set up a new module for each subsystem and function. In addition to these contributions, the first team also implemented all of the functions and math modules in Verilog and developed a method for progress tracking.

The goals of the second team were to assist in the completion of the implementation phase of the encoder module, develop a system level validation process, and interface the encoder to a CPU. The second team accomplished their first goal by helping the first team to implement the remaining encoder modules. The team used a combination of scripting and outsourcing work to a dedicated server to achieve an efficient validation process. Python scripts were used to ensure proper coding style and file system structure, while the timing analysis simulations were executed on a dedicated server to allow for faster results. The team's final goal was to interface the encoder with a CPU. The Xilinx Microblaze soft processor was implemented on a Xilinx Virtex-5 FPGA and the CODEC tied to the processor's main communication bus. The interface between the CODEC and the processor was built using memory mapped registers. This configuration allowed the CPU to send samples to the CODEC and retrieve the output bit stream for validation. In addition to completing these goals, the second team also established the use of a revision control system to maintain the codebase for the project.

The third team's goals were to adapt the architecture created by the first team to work with the decoder, and implement the decoder's subsystems. While the base architecture for the decoder is the same as the encoder, modifications to the memory modules and refactoring the input and output buffers to accept the opposite flow of data were necessary to implement the decoder. Along with the implementation of the decoder's modules, the third team also created a more effective work tracking system to use in progress meetings with advisors and NASA engineers.

The fourth team's goals were to assist in the completion of the implementation phase of the decoder module, interface the decoder to a CPU, and create a single integrated encoder and decoder prototype for functionality demonstration. The integrated encoder/decoder device records encoded voice samples from a microphone to on-board memory when the record button is pressed, and plays back decoded voice samples to speakers when the play button is pressed.

ANALYSIS

Inter-team Cooperation

Multi-team projects provide opportunities that individual teams do not. One of these opportunities is inter-team cooperation. In order for multiple teams to accomplish separate goals on the same project, it is vital that the teams work together seamlessly. For the CODEC project, the team sizes (2 to 5 members per team) and academic setting allowed for easy communication between teams. Also, the teams were able to have a reserved team workspace in which team members from all teams could come and go according to their schedules and work side-by-side with other team members. This atmosphere encouraged interaction between the teams, allowed team members to motivate each other to accomplish their goals, and also made troubleshooting problems easier because team members could discuss problems in real-time with other members who may have already encountered and solved the problem.

Project Management

Project management is another key component of capstone projects that provides extra experiences to members of multi-team projects. The most valuable of these experiences is learning how to adapt the project environment to support the multi-team dynamic. Initially, the first team was able to easily communicate and separate tasks with one another because the team only had two members. However, as the second team began, it was necessary to develop a more concrete system for distributing tasks among team members. Furthermore, scheduling became a more difficult challenge. For the most part, scheduling inter-team meetings was most easily accomplished by appointing one member from each team to coordinate with their respective teams and one another. This method, along with scheduling advisor meetings with all team members at the same time, created sufficient interaction between team members to accomplish the project goals.

Another experience provided by a multi-team project is working in an open-source development environment. With the introduction of the second team, a revision control system was necessary to maintain the code. The revision control system chosen for the CODEC project was Git-Hub.org [2]. Git-Hub provided the project with a way for the latest code to be stored where all team members could access it. This became particularly important when the project reached the phase where a portion of the team members were working on modules while other team members were performing timing analysis tests on recently competed modules. Another important part of open-source development is a task distribution system. For a project that has many exclusive tasks to be accomplished, it is imperative to have a system in place to track the tasks, who the tasks are assigned to, and the current status of the tasks. The CODEC teams eventually adopted a system that utilizes Microsoft Excel to outline the hierarchy of sub-modules (functions, math blocks, etc.) necessary to construct the encoder/decoder system. Keeping this file under revision control allowed team members to pull the latest task assignments and select the next task on the list after completing a task. Color coding was used to indicate the various statuses associated with each sub-module. Figure 3 shows a portion of the task list used by the encoder teams.

	A	B	C	D	E	F	G	H	I	J
59			Unstarted		In progress		Completed			
60	Function Location		Function Names							Current Assignment
61	filter.c		Syn_filt							Nick
62	pitch.c		Pitch_fr3							Troy
63	pitch.c			Norm_Corr						Troy
64	filter.c				Convolve					Nick
65	dspfunc.c				Inv_sqrt					Cooper
66	pitch.c			Interpol_3						Cooper
67	pitch.c		Enc_lag3							Troy
68	p_parity.c		Parity_Pitch							Sean
69	pred_lt3.c		Pred_lt_3							Troy
70	filter.c		Convolve							Nick
71	pitch.c		G_pitch							Sean
72	cod_ld8k.c		test_err							Sean
73	acelp_co.c		ACELP_Codebook							Zach
74	acelp_co.c			Cor_h						Zach
75	acelp_co.c			Cor_h_X						Zach
76	acelp_co.c			D4i40_17						Zach
77	cod_ld8k.c		Corr_xy2							Zach
78	qua_gain.c		Qua_gain							Sean
79	gainpred.c			Gain_predict						Sean
80	dspfunc.c				Log2					Sean
81	dspfunc.c				Pow2					Sean

Figure 3: Task List Used to Coordinate Module Design

An added benefit of this task tracking system is that it reduced the necessity for face-to-face communication. Team members could easily check the status of the project by accessing the task list.

A final component of the open-source model was the creation of a project wiki. The wiki was established as an aid for incoming teams to easily access knowledge gained by the previous teams and more quickly become familiar with the project. The wiki was hosted through the Git-Hub website and contained information such as module templates, architecture explanations, a troubleshooting section, and tutorials on how to perform validation of sub-modules.

Although the student management infrastructure took almost six months to fully develop, faculty advisor meetings were a requirement from the first day. These weekly progress updates were invaluable to the teams, not only to stay motivated to complete work regularly, but also to work out any road blocks encountered during the week that could not be easily solved by the team members. The advisor meetings began as a semi-informal meeting between the faculty advisor and the two members of the first team. However, when the transition to two teams was made, the advisor meetings became much more formal, and eventually developed a standard agenda. Each meeting began with updates from each member on his work over the previous week. This was followed by an open question-and-answer session to discuss semester goals and extra work associated with the requirements of the department’s capstone course requirements such as design documents and review presentations. At least once each semester, all members would meet to participate in a teleconference with the NASA engineers that were overseeing the project. These meetings were invaluable to the students, who gained experience interacting with working engineers as well as real clients with real specification requirements.

Curriculum Adherence

While there are many benefits to working on a multi-team capstone project, there are also detriments. One of the hardest challenges faced by members of the various CODEC teams was adhering to the requirements of the capstone course itself. Mississippi State University's Electrical and Computer Engineering Department's capstone course was designed to be a two-semester, one-team project; however, the CODEC project was a five-semester, four-team project. Furthermore, the course typically requires a physical prototype at the end of the first semester, and a working final package at the end of the second semester. These requirements could not be met for multiple reasons. The first reason is the nature of the project. Because the CODEC was being developed in Verilog HDL, there was no physical aspect to the project. This forced the teams to develop a software validation method provide results equivalent to both a prototype and final working package. The second reason the requirements could not be met was the size of the project. The Encoder sub-system by itself was comprised of over 50 Verilog modules and over 60,000 lines of code. Therefore, it was unreasonable to expect both the Encoder and Decoder systems be completed in a period of eight months, leading to the adoption of the four-team model. The decision to move to this larger, multi-team system was made early during the design cycle for the first team. It was initially determined that coding the entire encoder and decoder was too much work for two people. During the second team's planning phase, the decision to increase the number of teams from two to four was made. This decision was made to separate the FPGA implementation aspect of the project from the HDL coding section.

The new four-team model implied that the goals for each team would vary depending on the overall progress of the project. Each team was required to adhere to the guidelines of the course while still progressing towards the completion of the CODEC. To this end, each team performed the same design steps as any one-team capstone project. These steps can be generalized to the following: Identification of the problem; Proposition of a Solution; Implementation of the Solution; Validation of the Solution. For the first team, identification of the problem required the determination that an HDL system was a legitimate method by which size, weight and power could be reduced in voice communications. The proposed solution to this problem was to design an architecture in HDL that would support the ITU G.729 CODEC. The first team designed the proposed architecture and implemented the necessary sub-systems to create a functioning encoder. Finally, behavioral simulation was used to validate that the encoder system produced correct output. To perform these simulations, test vectors were run on the C-implementation to produce output values. These test vectors were then manually fed into the module to be tested and the output checked against the C-output values. The second team's problem identification was how to implement the encoder in a physical system. The proposed solution was to connect the encoder to the peripheral bus of a soft processor on an FPGA. The team used available tools to create a working system on the FPGA and validated it using one of the ITU provided test vectors. The problem faced by the third team was how to adapt the previously designed encoder architecture to implement the decoder system. The third team proposed minimal changes to the memory buffers to accommodate the difference and implemented the new architecture as well as the sub-modules required to build the decoder. Again, simulation was used to validate the results of the decoder system. The fourth team defined its problem as designing an integrated encoder and decoder device to serve as a prototype system for delivery to NASA. The fully functional prototype was required to sample voice from a line in input connected to an external microphone. The prototype required a means to store encoded voice samples and a mechanism to transfer input and output samples to a PC. Finally, the prototype was required to play back decoded samples to an external speaker. Removable flash media was chosen as the primary storage device in order to provide a simple means of transferring encoder and decoder input and output data to and from a PC. Voice samples captured from the microphone were stored on the removable media in encoded and pre-encoded formats. Decoded voice samples were sent to the external speakers and stored on the removable media. Storing both inputs and outputs to removable media supported external testing of CODEC functionality.

CONCLUSION

The CODEC project is an excellent example of a multi-team capstone project. It illustrates how students can gain invaluable experiences vital to success prior to entering the workforce. This will give students the ability to enter into a similar environment in industry and perform effectively immediately rather than spend extra time adapting to the new team mechanics.

Engineers in industry often work in teams across large geographic distances. Often geographically separated teams subdivide an integrated circuit design task into large functional blocks. This capstone project simulated this parallel design approach by breaking the ITU G.729 CODEC into separate encoder and decoder sub-modules. The encoder and decoder modules where also broken into multiple sub-modules which were implemented by separate design teams. Digital design engineers in industry also often work serially on chip design projects. Teams regularly inherit specifications, model-based design descriptions, and existing functional blocks from other teams. The staggered start dates of each of the four teams of this capstone project were similar to the serial work flows found in industry.

Not every capstone project has the size to support a multi-team staggered approach. However, there are many that would benefit greatly from it. Specifically, projects that require multiple backgrounds to complete independent tasks can benefit from distributing these tasks to multiple teams rather than combining these students onto a single team. The presented project's multi-team model and the staggered project start dates for this capstone project forced inter-team cooperation, project management, led students to move to an open-source development model, and unique and positive learning experience for fourteen electrical and computer engineering students. Using this model for capstone projects will give future students the possibility to experience the same benefits.

REFERENCES

[1] ITU-T (2007-01) (PDF). G.729: Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP). p. i. Retrieved 2010-08-28.

[2] *Github.org*. Web. <http://github.org>.

Sean Owens
Bachelor of Science, Computer Engineering, Mississippi State University, Mississippi State, MS
Team 1 Member – CODEC Project
Research – Methods for Dynamic Partial Reconfiguration of FPGAs

Tommy Morris
Assistant Professor, Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS
Team 1-4 Advisor
Research - Partial Reconfiguration of FPGAs, Industrial Control System Cybersecurity

Gregory Hall
Senior Systems Architect, National Aeronautics and Space and Administration, Johnson Space Center, Houston, TX
Project Sponsor from NASA
Research – Command and Data Handling for NASA Human Space Flight Systems